

DECLARATION

My name is Donald Eyles, the listed inventor of Application No. 07/891,934, filed May 29, 1992 and I have first hand knowledge of the following events.

This Declaration contains a brief chronology of activity related to the Timeliner sequencing language. In addition, it cites a number of documents that are relevant to Timeliner's history.

Apollo era

For Apollo, my job was to develop flight software for the lunar landing phase of the LM mission. In doing so I was a user of a simulation capability called "Astronaut" whose function was to provide simulated man-in-the-loop inputs during a simulation. This experience was in my mind as I began to develop timeliner.

Astronaut was a much less general and capable version of a sequencing language, capable of only one of the functions provided by Timeliner (i.e. simulating astronaut input). Technically, I believe the only thing that Timeliner has in common with the old Astronaut is the use of the word "when" to create a construct that waits for some condition.

The Timeliner implementation was a fresh start and none of its code or logic came from Astronaut.

About 1980

The first version of Timeliner was developed, using the HAL computer language, about 1980. It may have been as early as 1979 or as late as 1981.

The development of Timeliner responded to the inadequacy of the test initialization capability that we used during the Shuttle "integrated simulation" effort. I believe that Timeliner was used for some integrated simulation testing. After the integrated effort was concluded Timeliner was incorporated into the OFS20 simulation. The HAL version is still in use in descendants of OFS20.

Reference document:

User's Guide to the Timeline Language, Don Eyles, undated

About 1986

For the Aeroassist Flight Experiment project I created a version of Timeliner in the Fortran language about 1986. This version was almost a straight recoding of the HAL version, with some capabilities left out. For example, since AFE is an unmanned vehicle, the ability to input keystrokes was omitted.

The Fortran version used a somewhat different method of maintaining the "variable list" that provided the information

needed to access particular variables in the simulation. At the level of the essential Timeliner characteristics there was no difference.

Reference document:

User's Guide to the Timeliner Language, Don Eyles, December 18, 1986.

April - July 1990

In the spring of 1990 I have the opportunity to review the Specification that described the User Interface Language intended for use aboard the space station and in ground applications. I realized upon looking at this document that my perspectives gained while creating and using Timeliner would be of value to the effort to define UIL.

Several documents came out of this period. The first was a 45 page "comment" on the UIL Specification. This document mentioned Timeliner, and proposed modifying the constructs proposed for UIL in a way that would have made UIL more like Timeliner. This document was the subject of a small meeting at NASA space station headquarters in Reston, Virginia on June 12. This meeting was chaired by Peg Snyder of NASA, and included Randy Davis, a professor at the University of Colorado who was the principal designer of the UIL. The document was also discussed at a meeting in Houston in July attended by me, Davis, NASA people including Snyder, and other personnel who work for space station contractors including MacDonnel Douglas, IBM, and Lockheed.

These documents taken together disclosed only the existence of Timeliner to NASA and some of its contractors and no offer to provide Timeliner was made.

Reference documents:

Comments on UIL Specification (Revised), Don Eyles, May 14, 1990

Report on UIL Meeting at Reston, Don Eyles, June 21, 1990

How TIMELINER Works, Don Eyles, June 29, 1990

AUTUMN 1990

During the fall of 1990 I started building a version of Timeliner in the Ada language for use as a simulation sequencer in the "DMS testbed" project. The test-bed was (and is) an effort to help NASA asses the requirements and performance of the data management system (DMS) services needed to support the management of data aboard the space station.

A number of changes and new capabilities were added to the Ada version. Some of the changes were aimed at meeting requirements that the experience of reviewing UIL made me realize were useful. In my mind at this time (but only there), I saw Timeliner as a candidate to replace UIL.

The new features added to the Ada version at this time included:

- (1) The compile-time and execution-time functions of Timeliner were separated. Whereas before the "compilation" of a script was part of the initialization of each simulation, the new version compiled scripts as a separate step. Thus recompilation was not required when the same script was used repeatedly.
- (2) A new organizational level called the "bundle" was added to support the packaging of related sequences.
- (3) The optional OTHERWISE statement was added to the WHEN construct.
- (4) The name of the UNTIL statement was changed to BEFORE, and the name of the FOR statement was changed to WITHIN. These changes were aimed at making the constructs more English-like, especially when an OTHERWISE statement was also included (For example "When A before B do X, otherwise do Y" sounds better than "When A until B do X, otherwise do Y".)
- (5) Whereas the previous versions of Timeliner used pairs of DO and END statements to group statements within constructs when necessary, the Ada version included explicit END statements to conclude each construct.
- (6) The WAIT statement was simplified to remove capability that had become redundant.

The Ada version was documented in the form of a user's guide for Draper personnel. Appendix A of the guide summarized the changes made in the Ada version.

Reference document:

User's Guide to the Timeliner Language, Ada-Language Version, Don Eyles and Concetta Cuevas, March 6, 1991.

May 1991 February 1992

In May 1991 Rick Eckelkamp of NASA called a meeting to discussed alternatives to the UIL language. This initiative was caused by the fact that the development of UIL had been postponed and it would not be available for the early space station missions.

I traveled to Houston and made a presentation to NASA personnel that described Timeliner and offered it as an alternative. Timeliner was well received. I made a number of presentations on the subject of Timeliner during this period.

Another document from this period is a paper written for, and presented at the AIAA Computing in Aerospace conference in Baltimore in October, 1991. This document represents the first public disclosure of the idea behind Timeliner.

At a meeting in September 1991 of the Program Software Control Board (PSCB), Richard Thorsen verbally approved Timeliner for use aboard the station.

In October 1991 we started preparing a new UIL Specification document based on Timeliner, which is included below.

With respect to the existing Ada version, the version described in this Specification included changes in the user-interface commands available for controlling Timeliner's operation in real time, and changes to the "action" statements required to interface with the target system, in this case the space station.

The version of the Specification included is the latest "clean version to be produced, at Charles Stark Draper Laboratory, Inc, in February 1992. Since then the document has been further modified by NASA, mainly to make it applicable to a wider range of applications. The decision to officially incorporate this version into the program was expected in March or April 1992.

Reference documents:

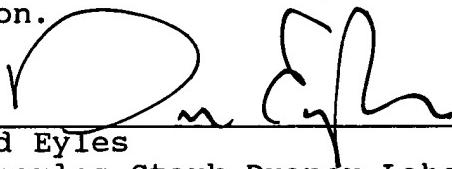
An Introduction To Timeliner, Don Eyles, May 24, 1991.

A Time-Oriented Language for the Writing of Procedures to Sequence the Operation of a Spacecraft and Its Systems, Don Eyles, June 28, 1991 (NASA approval data). This paper carries a Draper copyright notice.

An Introduction To Timeliner, Don Eyles, February 1992.

Specification, User Interface Language, February 1992.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and that like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.



Donald Eyles
The Charles Stark Draper Laboratory, Inc.
Mail Station #31
555 Technology Square
Cambridge, MA 02139

4/12/94

Date: